A Parametric Augmented Lagrangian Algorithm for Real-Time Economic NMPC

Jean-Hubert Hours, Harsh Shukla and Colin N. Jones

Abstract— **In this paper, a novel optimality-tracking algorithm for solving Economic Nonlinear Model Predictive Control (ENMPC) problems in real-time is presented. Developing online schemes for ENMPC is challenging, since it is unclear how convexity of the Quadratic Programming (QP) problem, which is obtained by linearisation of the NMPC program around the current iterate, can be enforced efficiently. Therefore, we propose addressing the problem by means of an augmented Lagrangian formulation. Our tracking scheme consists of a fixed number of inexact Newton steps computed on an augmented Lagrangian subproblem followed by a dual update per time step. Under mild assumptions on the number of iterations and the penalty parameter, it can be proven that the sub-optimality error provided by the parametric algorithm remains bounded over time. This result extends the authors' previous works from a theoretical and a computational perspective. Efficacy of the approach is demonstrated on an ENMPC example consisting of a bioreactor.**

I. INTRODUCTION

In NMPC, a control input is computed as a solution of a parametric Optimal Control Problem (pOCP) at each sampling time. Solving a pOCP at a high sampling rate is not computationally tractable and may result in unacceptable latency when dealing with unstable systems. Therefore, a significant research effort has been made to alleviate the computational burden of NMPC e.g. [6,7,17,22,23]. An important class of real-time optimisation techniques applicable to online NMPC are continuation methods [1]. They exploit the parametric nature of the NMPC problem by computing predictor and corrector steps along the state trajectory. Among such techniques, the real-time iteration scheme of [6] has proven efficient on NMPC problems with a least-squares stage-cost [13]. Such NMPC programs generally appear when the control objective is set-point stabilisation or trajectory tracking. The online strategy of [6] consists in solving a single parametric convex QP per time step. Convexity of the QP is enforced by means of the Gauss-Newton approximation [16], which provides an accurate hessian estimate when the residuals are small.

When it comes to ENMPC, the stage-cost corresponds to an economic performance index. Besides theoretical challenges regarding Lyapunov stability of the closed-loop system under the ENMPC control law [6], it appears to be difficult to develop an efficient real-time ENMPC algorithm. As the stage-cost does not have a least-squares structure, the Gauss-Newton hessian estimate is not applicable. To tackle this issue, a novel SQP-based real-time strategy was proposed in [19]. In order to obtain a convex QP, a mirrored hessian is used, which is made positive semidefinite by clipping the negative eigenvalues of the hessian of the Lagrangian. Despite its computational efficiency, the approach of [19] does not come with theoretical stability properties.

In this paper, we propose a novel real-time approach to ENMPC. Instead of constructing a local convex QP approximation of the Nonlinear Program (NLP) resulting from the discretisation of the pOCP as in [6,19], we form an augmented Lagrangian relaxation of the ENMPC problem [16]. The parametric augmented Lagrangian program is bound-constrained. Hence, a gradient projection can be computed efficiently and provides an active-set. A Newton subproblem is then solved inexactly on the current active-set. Global convergence of the process is guaranteed by means of a trust region [5]. The real-time scheme consists of computing a fixed number of trust region iterations on the augmented Lagrangian and then performing a first-order update of the Lagrange multiplier. Under mild assumptions on the ENMPC problem, it is shown that the suboptimal solution provided by our real-time scheme remains around an optimal solution of the ENMPC problem. Thus, the optimality-tracking scheme is stable under some conditions on the penalty coefficient and the number of primal iterations. Such a stability result for inexact Newton iterations on a parametric augmented Lagrangian is novel and did not appear in the authors' previous works [10,12]. In particular, it is based a local convergence rate for trust region Newton methods that does not rely on a finite detection of an optimal active-set. This is novel compared to the trust region literature [4].

In Section III, the multiple-shooting discretisation of the economic pOCP is presented and our real-time augmented Lagrangian algorithm is introduced. In Section IV, the stability properties of the scheme are analysed. In particular, we establish an inequality, which shows that the optimality-tracking method is contractive under some requirements on the penalty and the number of primal iterations. Finally, in Section V, our approach is demonstrated by applying ENMPC to control a bioreactor in order to maximise its productivity.

The work leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Unions Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no 607957 (TEMPO) and ERC grant agreement no. 307608 (BuildNet). All three authors are with the Automatic Control Laboratory, EPFL, Switzerland. jean-hubert.hours,harsh.shukla,colin.jones@epfl.ch

II. NOTATION

The indicator function of a closed subset Ω in $\mathbb{R}^n$ is denoted by ι_Ω and is defined as

$$\iota_\Omega(x) = \begin{cases} 0 & \text{if } x \in \Omega \\ +\infty & \text{if } x \notin \Omega \end{cases} .$$

The open ball with center x and radius r is denoted by $\mathcal{B}(x, r)$.

III. A REAL-TIME ENMPC SCHEME

A. Problem formulation

We address the problem of computing a first-order solution $(x^*(\cdot), u^*(\cdot))$ to the continuous-time economic NMPC problem

$$\begin{aligned} &\underset{x(\cdot),u(\cdot)}{\text{minimise}} \int_0^T l(x(t), u(t))\,\mathrm{d}t \\ &\text{s.t. } x(0) = \hat{x}_0 \ , \\ &\forall t \in [0,T],\ \dot{x}(t) = f(x(t), u(t)) \ , \\ &\forall t \in [0,T],\ \underline{x} \le x(t) \le \bar{x},\ \underline{u} \le u(t) \le \bar{u} \ , \end{aligned} \tag{1}$$

for an initial condition $\hat{x}_0 \in \mathbb{R}^{n_x}$, a prediction horizon T and a continuously differentiable stage-cost $l : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$. The function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is assumed to be continuously differentiable on the set $\Omega := \mathcal{X} \times \mathcal{U}$, which is defined by

$$\begin{aligned} \mathcal{X} &:= \{x \in \mathbb{R}^{n_x} \ : \ \underline{x} \le x \le \bar{x}\} \ , \\ \mathcal{U} &:= \{u \in \mathbb{R}^{n_u} \ : \ \underline{u} \le u \le \bar{u}\} \ . \end{aligned}$$

The pOCP (1) is to be solved in real-time for varying initial conditions $\hat{x}_0$.

Remark 3.1: The terminal cost and constraint set, which are required to ensure closed-stability of the nonlinear system under the closed-loop optimal ENMPC solution u^*, are neglected. Closed-loop asymptotic stability under an economic NMPC control law has been investigated by [8] under some dissipativity and controllability assumptions.

In order to transform the pOCP (1) into a finite-dimensional NLP, direct methods proceed by parameterising the continuous control profile $u(\cdot)$ using a finite number of parameters, whose optimal values can be computed by means of a nonlinear solver. Similarly to [15], we resort to a piecewise constant parameterisation of the control profile, which is written as follows, given a time instant $t \in [0, T]$,

$$u(t) = \sum_{i=0}^{N-1} q_i \iota_{[t_i, t_{i+1}]}(t) \ ,$$

where $N \ge 1$, $\{q_i\}_0^{N-1} \subset \mathbb{R}^{n_u}$ and the mesh $\{t_i\}_{i=0}^N \subset [0,T]$ is such that $t_0 := 0$ and $t_N := T$. In order to parameterise the state profile $x(\cdot)$, shooting nodes $\{s_i\}_{i=0}^N \subset \mathbb{R}^{n_x}$ and shooting constraints are introduced for each interval $[t_i, t_{i+1}]$ as follows

$$s_{i+1} - x(t_{i+1}; s_i, q_i) = 0,\ i \in \{0, N-1\} \ , \tag{2}$$

where $x(t_{i+1}; s_i, q_i)$ is the solution of the boundary value problem

$$\begin{cases} \forall t \in [t_i, t_{i+1}],\ \dot{x}(t) = f(x(t), q_i) \\ x(t_i) = s_i \ . \end{cases} \tag{3}$$

The role of the shooting constraints is to ensure continuity of the state profile at the ends of every shooting interval. The objective of the pOCP (1) is also subdivided according to the mesh $\{t_i\}_{i=0}^N$ as follows

$$\begin{aligned} \int_0^T l(x(t), u(t))\,\mathrm{d}t &= \sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} l(x(t), u(t))\,\mathrm{d}t \\ &= \sum_{i=0}^{N-1} \omega(t_{i+1}; s_i, q_i) \ , \end{aligned}$$

where $\omega(t_{i+1}; s_i, q_i)$ is the solution of the boundary value problem

$$\begin{cases} \forall t \in [t_i, t_{i+1}],\ \dot{y}(t; s_i, q_i) = l(x(t; s_i, q_i), q_i) \\ \omega(t_i; s_i, q_i) = 0 \ , \end{cases} \tag{4}$$

which is coupled with (3) via the state $x(t; s_i, q_i)$. Finally, the paramteric NLP resulting from the multiple-shooting discretisation is

$$\begin{aligned} \mathcal{P}(\hat{x}_0) : \underset{\{s_i\}_{i=0}^N, \{q_i\}_{i=0}^{N-1}}{\text{minimise}} \ & \sum_{i=0}^{N-1} \omega(t_{i+1}; s_i, q_i) \\ \text{s.t. } & s_0 - \hat{x}_0 = 0, \\ & s_{i+1} - x(t_{i+1}; s_i, q_i) = 0, \\ & s_i \in \mathcal{X},\ q_i \in \mathcal{U},\ i \in \{0, \dots, N-1\} \ , \end{aligned} \tag{5}$$

where for all $i \in \{0, \dots, N-1\}$, $x(t_{i+1}; s_i, q_i)$ and $\omega(t_{i+1}; s_i, q_i)$ are solutions of the following augmented boundary value problem

$$\begin{cases} \forall t \in [t_i, t_{i+1}],\ \dot{v}(t; s_i, q_i) = F(v(t; s_i, q_i), q_i) \\ v(t_i; s_i, q_i) = \begin{pmatrix} s_i \\ 0 \end{pmatrix} \ . \end{cases} \tag{6}$$

where the mapping $F : \mathbb{R}^{n_x+1} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x+1}$

$$F(v(t; s_i, q_i), q_i) := \begin{pmatrix} f(x(t; s_i, q_i), q_i) \\ l(x(t; s_i, q_i), q_i) \end{pmatrix} \ ,$$

and the augmented state is denoted by

$$v(t; s_i, q_i) := \begin{pmatrix} x(t; s_i, q_i) \\ \omega(t; s_i, q_i) \end{pmatrix} \in \mathbb{R}^{n_x+1} \ .$$

B. Algorithm description

Instead of linearising the shooting constraint (2) with respect to the shooting node s_i and control q_i as in [6,13,15], we relax it by means of an augmented Lagrangian penalty and thus introduce

$$\begin{aligned} \mathcal{L}_\varrho(z, \mu, \hat{x}_0) := & \left(\mu_0 + \frac{\varrho}{2}(s_0 - \hat{x}_0)\right)^\top (s_0 - \hat{x}_0) \\ & + \sum_{i=1}^{N} L_\varrho(v(t_i; s_{i-1}, q_{i-1}), s_i, \mu_i) \ , \end{aligned} \tag{7}$$

where the local augmented Lagrangian is defined by

$$L_\varrho\left(v\left(t;s,q\right),s',\nu\right) := \omega\left(t;s,q\right) + \left(\nu + \frac{\varrho}{2}\left(s' - x\left(t;s,q\right)\right)\right)^\top \left(s' - x\left(t;s,q\right)\right) \ ,$$

for $s, s' \in \mathbb{R}^{n_x}$, $q \in \mathbb{R}^{n_u}$, a Lagrange multiplier $\nu \in \mathbb{R}^{n_x}$, a penalty $\varrho > 0$ and

$$z := \left(s_0^\top, q_0^\top, \ldots, s_{N-1}^\top, q_{N-1}^\top, s_N^\top\right)^\top \in \mathbb{R}^n$$
$$\mu := \left(\mu_0^\top, \ldots, \mu_N^\top\right)^\top \in \mathbb{R}^{n_x(N+1)} \ ,$$

with $n := N\left(n_x + n_u\right) + n_x$. Our real-time scheme then proceeds by minimising the augmented Lagrangian (7) and updates the Lagrange multiplier μ only once per time instant while the state evolves from $\hat{x}_0$ to $\tilde{x}_0$, as described in Algorithm 1 below.

Algorithm 1 Optimality-tracking descent-based algorithm

Input: Suboptimal point $\left(\bar{z}\left(\hat{x}_0\right)^\top, \bar{\mu}\left(\hat{x}_0\right)^\top\right)^\top$, state $\tilde{x}_0$ and augmented Lagrangian function $\mathcal{L}_\varrho\left(\cdot, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right) + \iota_\Omega$.
Descent phase: Apply M iterations of Algorithm 2 initialised at $\bar{z}\left(\hat{x}_0\right)$ to minimise the augmented Lagrangian $\mathcal{L}_\varrho\left(\cdot, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right) + \iota_\Omega$ and obtain a suboptimal primal iterate z^M.
$\bar{z}\left(\tilde{x}_0\right) \leftarrow z^M$
Dual update: $\bar{\mu}\left(\tilde{x}_0\right) \leftarrow \bar{\mu}\left(\hat{x}_0\right) + \varrho G\left(\bar{z}\left(\tilde{x}_0\right), \tilde{x}_0\right)$
Output: Primal-dual point $\left(\bar{z}\left(\tilde{x}_0\right)^\top, \bar{\mu}\left(\tilde{x}_0\right)^\top\right)^\top$.

An efficient strategy to compute an approximate critical point of the augmented Lagrangian (7) is the truncated trust region Newton method described in Algorithm 2 below. At every iteration of Algorithm 2, a Newton model

Algorithm 2 Trust region algorithm on the parametric augmented Lagrangian

1: **Constants:** Initial trust region radius Δ, coefficient $\gamma_2 > 0$ and regularisation coefficient r.
2: **Input:** Suboptimal primal variable $\bar{z}\left(\hat{x}_0\right)$ and objective function $\mathcal{L}_\varrho\left(\cdot, \bar{\mu}\left(\tilde{x}_0\right), \tilde{x}_0\right) + \iota_\Omega$.
3: **Warm-start:** $z \leftarrow \bar{z}\left(\hat{x}_0\right)$
4: **for** $l = 1, \ldots, M$ **do**
5: **Active set computation**
6: Compute the Cauchy point z_c as in Eq. (9) according
7: to the requirements in (10).
8: **Refinement phase**
9: Find $y \in \Omega$ by approximately solving (11) via sCG
10: iterations initialised at z_c [21].
11: **Trust-region update**
12: Compute the test ratio ρ according to Eq. (13).
13: Shrink or increase the trust region radius Δ
14: and update z depending on the ratio ρ.
15: **end for**
16: **Output:** $z^M = z$

$m_\varrho\left(\cdot, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right)$ of the augmented Lagrangian (7) is constructed around the current iterate z as follows

$$m_\varrho\left(y, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right) := \mathcal{L}_\varrho\left(z, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right) + \nabla_z \mathcal{L}_\varrho\left(z, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right)^\top \left(y - z\right) + \frac{1}{2}\left(y - z\right)^\top \nabla^2_{z,z} \mathcal{L}_\varrho\left(z, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right)\left(y - z\right) \ . \tag{8}$$

An active-set is then computed by means of a gradient projection. More precisely, it is the set of active constraints $\mathcal{A}_\Omega\left(z_c\right)$ at the Cauchy point

$$z_c := P_\Omega\left(z - \alpha \nabla_z \mathcal{L}_\varrho\left(z, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right)\right) \ , \tag{9}$$

where the step-size α is such that

$$\begin{cases} m_\varrho\left(z_c, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right) \leq m_\varrho\left(z, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right) \\ \qquad\qquad + \nu_0 \nabla_z \mathcal{L}\left(z, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right)^\top \left(z_c - z\right) \\ \left\|z_c - z\right\|_\infty \leq \nu_1 \Delta \ . \end{cases} \tag{10}$$

where $\nu_0 \in (0,1)$ and $\nu_1 > 0$. For clarity, we omit other standard conditions on α, which can be found in [4]. The first condition in Eq. (10) corresponds to a sufficient decrease of the model function, while the second condition ensures that the Cauchy point lies in a scaled trust region around the iterate z. A point that fulfils the requirements in (10) can be computed by means of a projected search [5]. Once an active-set has been computed by means of the Cauchy point z_c, the model function is further minimised on the subspace of free variables at z_c. A candidate point y in Ω is obtained by approximately solving the nonconvex QP

$$\begin{aligned} \underset{y \in \Omega}{\text{minimise}} \ & m_\varrho\left(y, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right) + \frac{r}{2}\left\|y - z_c\right\|_2^2 \\ \text{s.t.} \ & \left\|y - z\right\|_\infty \leq \gamma_2 \Delta \ , \\ & \mathcal{A}_\Omega\left(z_c\right) \subseteq \mathcal{A}_\Omega\left(y\right) \ , \end{aligned} \tag{11}$$

where r is a regularisation coefficient. The proximal regularisation in problem (11) is the main novelty of Algorithm 2 compared to standard trust region methods. The point y is yielded by safeguarded Conjugate Gradient (sCG) iterations, which are aborted if a negative curvature direction is encountered or a problem bound is hit during search [21]. An important point is that the sCG procedure is to be initialised at the Cauchy point z_c. As sCG guarantees decrease of the regularised model function at each iteration [21], the proximal regularisation allows for a quantification of the decrease provided by the refinement phase, namely

$$m_\varrho\left(y, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right) + \frac{r}{2}\left\|y - z_c\right\|_2^2 \leq m_\varrho\left(z_c, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right) \ . \tag{12}$$

This is a key ingredient in the convergence analysis as explained in Section IV. Moreover, it does not hamper the fast local convergence property of Newton's method if the regularisation coefficient r is sufficiently small, as shown in [11]. In order to assess the quality of the candidate point y, the ratio

$$\rho := \frac{\mathcal{L}_\varrho\left(z, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right) - \mathcal{L}_\varrho\left(y, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right)}{m_\varrho\left(z, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right) - m_\varrho\left(y, \bar{\mu}\left(\hat{x}_0\right), \tilde{x}_0\right)} \tag{13}$$

is computed. If the ratio is below a threshold $\eta_1 \in (0,1)$, the trust region iteration is deemed to be unsuccessful and the trust region radius is shrunk. In case the ratio is above η_1, the iteration is considered as successful, the trust region radius is enlarged and the iterate z is updated to the candidate point y. This process is iterated M times to provide a suboptimal primal point $\bar{z}(\tilde{x}_0) := z^M$.

IV. STABILITY GUARANTEES

Algorithm 1 provides a suboptimal primal-dual point

$$\bar{w}(\tilde{x}_0) := \left(\bar{z}(\tilde{x}_0)^\top, \bar{\mu}(\tilde{x}_0)^\top\right)^\top \quad .$$

As Algorithm 1 is a truncated procedure warm-started at $\bar{w}(\hat{x}_0)$, it cannot be guaranteed to remain close to a first-order critical point of $\mathcal{P}(\tilde{x}_0)$

$$w^*(\tilde{x}_0) := \left(z^*(\tilde{x}_0)^\top, \mu^*(\tilde{x}_0)^\top\right)^\top \quad ,$$

if the state $\tilde{x}_0$ moves too far from the state $\hat{x}_0$. However, one can derive conditions on the parameter difference $\|\hat{x}_0 - \tilde{x}_0\|_2$, the number of iterations M and the penalty ϱ so that the optimality-tracking error

$$\epsilon(x) := \|\bar{w}(x) - w^*(x)\|_2 \tag{14}$$

remains within a bounded interval as the system's state x evolves. In this section, the closed-loop state trajectory under the suboptimal control law $\bar{q}_0(x)$ is sampled at discrete time instants, so that we can consider a state sequence $\{x_k\}$. To show that the error sequence $\{\epsilon(x_k)\}$ is stable, we require two ingredients, which are a local convergence rate for Algorithm 2 and some regularity properties of critical points of $\mathcal{P}(x)$ with respect to parameter changes. These elements translate into two fundamental properties of the parametric NLP $\mathcal{P}(x)$, namely the *Kurdyka-Lojasiewicz (KL) inequality* [2] and *Robinson's strong regularity* [20] respectively. A detailed formulation of these two properties in the context of NMPC can be found in the authors' previous work [10,12], thus we only give an overview of the theoretical results in the following two paragraphs.

A. Convergence rate of the trust region iterates

Using the arguments of [4], if M is infinite, it can be shown that the primal sequence generated by Algorithm 2 converges to a critical point $z^\infty(\bar{\mu}(\hat{x}_0), \tilde{x}_0)$ of the augmented Lagrangian (7). However, the asymptotically superlinear convergence rate of trust region Newton methods such as Algorithm 2 holds only once the active-set has settled down [4]. Therefore, resorting to such a convergence rate for our analysis does not seem to be appropriate, as the primal warm-start $\bar{z}(\hat{x}_0)$ cannot be assumed to lie on the same face of Ω as the critical point $z^\infty(\bar{\mu}(\hat{x}_0), \tilde{x}_0)$, especially if $\tilde{x}_0$ is far from $\hat{x}_0$. Therefore, we establish an asymptotic convergence rate for Algorithm 2, which does not rely on finite activity detection. The KL property turns out to be a key ingredient to obtain this result.

Assumption 4.1 (Polynomial data): The stage-cost l and the dynamics right hand side f are assumed to be polynomial functions.

Assumption 4.1 guarantees that the KL inequality is satisfied by the augmented Lagrangian (7), as it becomes a multivariate polynomial function [3]. Its degree is denoted by d_L. In plain words, the KL inequality corresponds to the fact that the norm of any subgradient grows faster than a power of the objective locally. Further details can be found in [2,3]. Based on the properties of the gradient projection onto the convex set Ω and inequality (12), it can be easily shown that a successful iteration of Algorithm 2 ensures a sufficient decrease of the augmented Lagrangian

$$\begin{aligned}\mathcal{L}_\varrho(z, \bar{\mu}(\hat{x}_0), \tilde{x}_0) - \mathcal{L}_\varrho(y, \bar{\mu}(\hat{x}_0), \tilde{x}_0) \\ \geq \kappa_1 \max\left\{\|z_c - z\|_2^2, \|y - z\|_2^2\right\} \quad ,\end{aligned} \tag{15}$$

where κ_1 is a positive constant. The sufficient decrease inequality (15), when combined with the KL property, plays an important role in the proof of Theorem (4.1) below.

Theorem 4.1 (Local convergence rate of Algorithm 2): Let $\{z^l\}$ be the sequence generated by Algorithm 2. There exists a radius $\eta > 0$ and a constant $C > 0$ such that if $\bar{z}(\hat{x}_0) \in \mathcal{B}(z^\infty(\bar{\mu}(\hat{x}_0), \tilde{x}_0), \eta)$,

$$\begin{aligned}\left\|z^M - z^\infty(\bar{\mu}(\hat{x}_0), \tilde{x}_0)\right\|_2 \leq \\ C\mathcal{S}(M)^{-\psi(d_L,n)} \|\bar{z}(\hat{x}_0) - z^\infty(\bar{\mu}(\hat{x}_0), \tilde{x}_0)\|_2 \quad ,\end{aligned} \tag{16}$$

where $\mathcal{S}(M)$ is the number of successful iterations from 0 to M and

$$\psi(d,n) := \frac{1}{d(3d-3)^{n-1} - 2} \quad .$$

The proof of Theorem 4.1 is omitted for brevity. It mainly follows the arguments of [2] after some changes required by the trust region procedure.

B. Local primal-dual contraction

As the state x_k changes, strong regularity of critical points of $\mathcal{P}(x_k)$ is assumed at all time instants k. In plain words, this property ensures Lipschitz continuity of critical points corresponding to different parameters, if the parameter difference is sufficiently small [20]. An important points is that it takes active-set changes into account.

Assumption 4.2 (Strong regularity at all points): For all $k \geq 0$, the associated critical points $w^*(x_k)$ of $\mathcal{P}(x_k)$ are strongly regular.

Under Assumption 4.2, a sequence of critical points $\{w^*(x_k)\}$ can be defined by means of Theorem 1 in [12]. The behaviour of the error sequence $\{\epsilon(x_k)\}$ between two consecutive time instants is then characterised by Theorem 4.2 below.

Theorem 4.2 (Primal-dual contraction): Assume that the suboptimal primal point $\bar{z}(x_{k+1})$ has been generated by Algorithm 2. Given a time index k, if the error $\epsilon(x_k)$, the number of primal iterations M, the penalty parameter ϱ and the parameter difference $\|x_{k+1} - x_k\|_2$ satisfy the conditions of Theorem 4 in [12], then the following inequality is satisfied for all time indices $k \geq 0$:

$$\epsilon(x_{k+1}) \leq \beta_w(\varrho, M)\,\epsilon(x_k) + \beta_s(\varrho, M)\,\|x_{k+1} - x_k\|_2 \quad ,$$

with

$$\beta_w(\varrho, M) := C(1+\varrho\lambda_G)\left(1+\frac{\lambda_B\lambda_H}{\varrho}\right)\mathcal{S}(M)^{-\psi(d_L,n)} + \frac{\lambda_B\lambda_H}{\varrho} \quad , \tag{17}$$

and

$$\beta_s(\varrho, M) := C(1+\varrho\lambda_G)\lambda_B\lambda_H\mathcal{S}(M)^{-\psi(d_L,n)} + \frac{\lambda_B\lambda_H\lambda_A\lambda_F}{\varrho} \quad , \tag{18}$$

where the positive constants $\lambda_A, \lambda_B, \lambda_F$ and λ_H are defined in [12].

In particular, the result of Theorem 4.2 holds if the parameter difference $\|x_{k+1} - x_k\|_2$ is sufficiently small to be in the neighbourhood defined by the strong regularity of $\mathcal{P}(x_k)$ [12,20]. In the context of NMPC, this is equivalent to saying that the sampling period has to be small in comparison to speed of the system's dynamics.

C. Stability of the error sequence

Theorem 4.2 shows that some conditions on the number of successful iterations $\mathcal{S}(M)$ and the penalty coefficient ϱ must be satisfied for the error sequence $\{\epsilon(x_k)\}$ generated by Algorithm 1 to remain within a bounded interval. In particular, the coefficients $\beta_w(\varrho, M)$ and $\beta_s(\varrho, M)$ should be strictly smaller than 1. Hence, the number of successful iterations in Algorithm 2 is to be sufficiently large to make the first terms in (17) and (18) small. At the same time, given a fixed number of successful iterations $\mathcal{S}(M)$, the penalty coefficient ϱ has to be above a specific threshold to ensure that the second summands in (17) and (18) do not prevent β_w and β_s from being below one. This shows that a trade-off between $\mathcal{S}(M)$ and ϱ needs to be found in order to guarantee stability of the error sequence $\{\epsilon(x_k)\}$.

Corollary 4.1 (Error stability): Assume that $\mathcal{S}(M)$ and ϱ are such that $\beta_w(\varrho, M)$ and $\beta_s(\varrho, M)$ are below one. Let r_ϵ be a positive scalar satisfying the conditions of Corollary 2 in [12]. If $\epsilon(x_0) < r_\epsilon$ and the state difference $\|x_{k+1} - x_k\|_2$ meets the requirement of Corollary 2 in [12], then for all $k \geq 0$,

$$\epsilon(x_k) < r_\epsilon \quad .$$

In plain words, Corollary 4.1 shows that if the optimality-tracking process is initialised in a sufficiently small neighbourhood of a critical point of $\mathcal{P}(x_0)$, the suboptimal sequence generated by Algorithm 2 evolves in a tube around an optimal trajectory, under mild conditions on the state difference, the number of successful iterations and the penalty coefficient. We refer to [9] for implementation details and computational considerations for the discussed algorithms.

V. NUMERICAL EXAMPLE

In this section, we consider a nonlinear continuous-time model of bioreactor for culture fermentation. The system has five states and one input. The system dynamics is provided in Eq. (19).

$$\dot{x}_1 = -Dx_1 + \mu(x)\,x_1 \tag{19a}$$

$$\dot{x}_2 = D(u - x_2) - \frac{\mu(x)\,x_1}{Y_{xs}} \tag{19b}$$

$$\dot{x}_3 = -Dx_3 + (\alpha\mu(x) + \beta)\,x_1 \tag{19c}$$

$$\dot{x}_4 = \frac{u}{T} \tag{19d}$$

$$\dot{x}_5 = \frac{x_1}{T} \tag{19e}$$

where

$$\mu(x) = \mu_m \frac{\left(1 - \frac{x_3}{P_m}\right)x_2}{K_m + x_2 + \frac{x_2^2}{K_i}} \quad .$$

For the sake of brevity, we do not discuss details about the model and numerical values of parameters and refer to [19]. We consider the following economic stage-cost

$$l(x, u) = \frac{-Dx_3}{T} \quad .$$

The input is subject to lower and upper bound,

$$\underline{u} = 28.7 \text{ g/L} \leq u \leq \bar{u} = 40.0 \text{ g/L} \quad .$$

The control objective is to maximise the average productivity. It is known that for system (19), the maximum productivity is obtained when operating in periodic mode [18]. Therefore, we enforce periodicity constraints in the ENMPC problem, as follows

$$x(0) = x(T) \quad .$$

Algorithm 1 has been implemented in C++ and tested on a processor with 2.5 GHz and 8 GB of RAM. For this simulation, the maximum number of trust region M was set to 3. The penalty ϱ was set to 100 and a banded preconditioner with 10 bands was used. In the ENMPC problem (1), a prediction horizon of $T = 48$ hours with $N = 20$ shooting intervals was set. In order to simulate the system, we applied a fourth-order explicit Runge Kutta scheme (RK4) with 20 steps. The evaluation of the shooting constraints in (1) was performed with 1 step of RK4. The suboptimal input

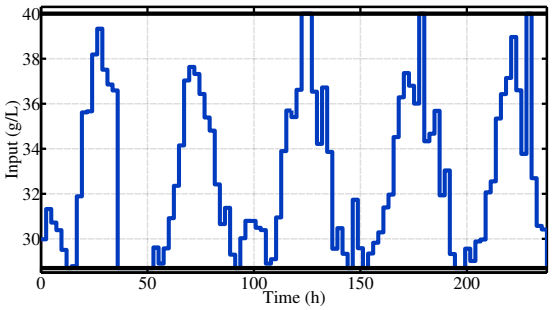

Fig. 1: Suboptimal input.

computed by our algorithm is shown in Fig 1

An average productivity of $3.08\ {}^{g}/_{L\cdot h}$ is obtained, which is a bit lower than the productivity given by the periodic trajectory proposed in [14] ($3.11\ {}^{g}/_{L\cdot h}$), but larger than the steady-state productivity ($3.0\ {}^{g}/_{L\cdot h}$). It is worth noting that the system operates in almost periodic mode under our suboptimal ENMPC control law.

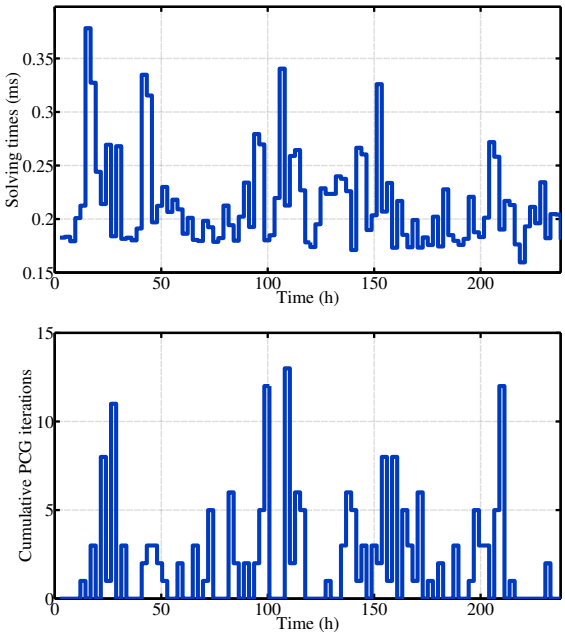

Fig. 2: Solving times and cumulative PCG iterations.

Computational results are shown in Fig. 2. An average solving time of 213μs was obtained. It appears that the cumulative number of sCG iterations per time step is quite low, which is a result of our preconditioning. This is interesting, as sCG iterations are one of the main computational burdens in Algorithm 1.

VI. CONCLUSIONS

A novel real-time algorithm applicable to economic NMPC has been presented. It has been shown that stability of the suboptimality error is guaranteed under mild assumptions on the algorithm parameters. The computational complexity of the algorithm has been analysed and it has been shown to scale favourably with the horizon length and the state and input dimensions. Finally, efficacy of the scheme has been demonstrated via real-time ENMPC on a bioreactor.

REFERENCES

[1] Allgower, E.L. and Georg, K. *Introduction to numerical continuation methods*. SIAM, 1990.

[2] Attouch, H. and Bolte, J. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116:5–16, 2009.

[3] Bolte, J. and Daniilidis, A. and Lewis, A. The Lojasiewicz inequality for nonsmooth sub-analytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17:1205–1223, 2007.

[4] J.V. Burke, J.J. Moré, and G. Toraldo. Convergence properties of trust region methods for linear and convex constraints. *Mathematical Programming*, 47:305–336, 1990.

[5] Conn, A.R. and Gould, N.I.M. and Toint, P.L. *Trust Region Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[6] Diehl, M. and Bock, H.G. and Schloeder, J.P. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, 43(5):1714–1736, 2005.

[7] Graichen, K. and Kugi, A. Stability and incremental improvement of suboptimal MPC without terminal constraints. *IEEE Transactions on Automatic Control*, 2010(11):2576–2580, November 2010.

[8] Grüne, L. and Stieler, M. Asymptotic stability and transient optimality of economic MPC without terminal conditions. *Journal of Process Control*, 24(8):1187–1196, 2014.

[9] Jean-Hubert Hours. *Decomposition Strategies for Nonconvex Problems, a Parametric Approach*. PhD thesis, 2016.

[10] Hours, J.-H. and Jones, C.N. A parametric multi-convex splitting technique with application to real-time NMPC. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, pages 5052–5057, Los Angeles, CA, USA, December 2014.

[11] Hours, J.-H. and Jones, C.N. An alternating trust region algorithm for distributed linearly constrained nonlinear programs, application to the AC optimal power flow. Technical Report EPFL-REPORT-205056, EPFL, 2015.

[12] Hours, J.-H. and Jones, C.N. A parametric nonconvex decomposition algorithm for real-time and distributed NMPC. *IEEE Transactions on Automatic Control*, 61(2), February 2016. To appear.

[13] Houska, B. and Ferreau, H.J. and Diehl, M. An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica*, 47:2279–2285, 2011.

[14] Houska, B. and Logist, F. and Van Impe, J. and Diehl, M. Approximate robust optimization of time-periodic stationary states with application to biochemical processes. In *Proceedings of the joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai, P.R. China, December 2009.

[15] Leineweber, D.B. Analyse und Restrukturierung eines Verfahrens zur direkten Lösung von Optimal-Steuerungsproblemen. Master's thesis, Universität Heidelberg, 1995.

[16] Nocedal, J. and Wright, S. *Numerical optimization*. Springer, New-York, 2006.

[17] Nuno M.C. De Oliveira and Lorenz T. Biegler. An extension of newton-type algorithms for nonlinear process control. *Automatica*, 31(2):281 – 286, 1995.

[18] Parulekar, S.J. Analysis of forced periodic operations of continuous bioprocesses - single input variations. *Chemical Engineering Science*, 53(14):2481–2502, 1998.

[19] Quirynen, R. and Houska, B. and Vallerio, M. and Telen, D. and Logist, F. and Van Impe, J. and Diehl, M. Symmetric algorithmic differentiation based exact hessian SQP method and software for economic MPC. In *Proceedings of the 53rd Conference on Decision and Control*, pages 2752–2757, Los Angeles, CA, USA, December 2014.

[20] Robinson, S.M. Strongly regular generalised equations. *Mathematics of Operations Research*, 5(1):43–62, 1980.

[21] Steihaug, T. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.

[22] Zavala, V.M. and Anitescu, M. Real-time nonlinear optimization as a generalised equation. *SIAM Journal on Control and Optimization*, 48(8):5444–5467, 2010.

[23] Zavala, V.M. and Biegler, L.T. The advanced-step NMPC controller: optimality, stability and robustness. *Automatica*, 45(1):86–93, 2009.